```
# UCSC Extension
# DBDA.X409.(10) MySQL and Oracle Database for Developers and
Designers
# Assignment 4
# Cheng Fei
# 10/31/2022
#
# A dynamic stored procedure to process constraints using the cursor
in the studentdb database.

# Change the current database to studentdb.
USE studentdb;

# @_CREATE_PROCEDURE_
# A database procedure in Dynamic SQL.
# Drop FK, PK, UK constraints in all studentdb's tables and add them
back again.
DELIMITER $
DROP PROCEDURE IF EXISTS change_studentdb_constraints$

CREATE PROCEDURE change_studentdb_constraints()
BEGIN
    # Store constraint information.
    DECLARE v_table_name VARCHAR(100);
  DECLARE v_column_name VARCHAR(100);
  DECLARE v_key_name VARCHAR(100);
  DECLARE key_done INT DEFAULT 0;
  DECLARE v_referenced_table_name VARCHAR(100);
  DECLARE v_referenced_column_name VARCHAR(100);
  # SQL instructions.
    DECLARE v_drop_constraint VARCHAR(1000);
  # Create a cursor to loop over tables.
  DECLARE key_cursor CURSOR FOR
        SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME,
            REFERENCED_TABLE_NAME,
REFERENCED_COLUMN_NAME
    FROM information_schema.key_column_usage
    WHERE table_schema = 'studentdb'
    ORDER BY REFERENCED_TABLE_NAME DESC; # Sort PRIMARY KEY
after FOREIGN KEY.

    BEGIN
        # Exception handler to stop the table loop after all
tables are visited.
        DECLARE EXIT HANDLER FOR NOT FOUND SET key_done = 1;
    # Loop over all tables in the studentdb database.
    OPEN key_cursor;
        key_loop: LOOP
            # Fetch current table's name.
            FETCH key_cursor INTO v_table_name,
v_column_name, v_key_name,
                v_referenced_table_name,
v_referenced_column_name;
            # Exit condition.
```

```
                        IF key_done = 1 THEN
                                LEAVE key_loop;
                        END IF;
                        # Change constraints' name by constraints'
type.
                        IF v_key_name != 'PRIMARY' AND v_table_name
= 'score' THEN
                                # Drop old FOREIGN KEY.
                                SET @v_drop_constraint :=
concat('ALTER TABLE ',

         v_table_name,

         ' DROP FOREIGN KEY ',

         v_key_name);
                                PREPARE stmt FROM
@v_drop_constraint;
                                EXECUTE stmt;
                        END IF;
                        IF v_table_name != 'score' THEN
                                # Drop old PRIMARY KEY.
                SET @v_drop_constraint := concat('ALTER TABLE ',

         v_table_name,

         ' DROP PRIMARY KEY');
                                PREPARE stmt FROM
@v_drop_constraint;
                                EXECUTE stmt;
                        END IF;
                END LOOP;
        CLOSE key_cursor;
    END;

END$
DELIMITER ;

# @_CREATE_PROCEDURE_
# A database procedure in Dynamic SQL.
# Add FK, PK, UK constraints to all studentdb's tables.
DELIMITER $
DROP PROCEDURE IF EXISTS add_studentdb_constraints$

CREATE PROCEDURE add_studentdb_constraints()
BEGIN
        DECLARE v_sql VARCHAR(1000);
        BEGIN
                SET @v_sql := concat('ALTER TABLE score ',
                                                         'ADD
CONSTRAINT FK_score_event_id ',
                                                         'FOREIGN
KEY (event_id) ',
                                                         'REFERENCES
grade_event(event_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;

        SET @v_sql := concat('ALTER TABLE score ',
                                                         'ADD
CONSTRAINT FK_score_student_id ',
                                                         'FOREIGN
KEY (student_id) ',
                                                         'REFERENCES
student(student_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;

        SET @v_sql := concat('ALTER TABLE score ',
                                                         'ADD
CONSTRAINT PK_score_event_id ',
                                                         'PRIMARY
KEY (event_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;

        SET @v_sql := concat('ALTER TABLE score ',
                                                         'ADD
CONSTRAINT PK_score_student_id ',
                                                         'PRIMARY
KEY (student_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;

        SET @v_sql := concat('ALTER TABLE student ',
                                                         'ADD
CONSTRAINT PK_student_student_id ',
                                                         'PRIMARY
KEY (student_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;

        SET @v_sql := concat('ALTER TABLE grade_event ',
                                                         'ADD
CONSTRAINT PK_grade_event_event_id ',
                                                         'PRIMARY
KEY (event_id)');
        PREPARE stmt FROM @v_sql;
        EXECUTE stmt;
        END;
END$
DELIMITER ;

# Commit changes.
COMMIT;
```